01. Cooking Masterclass

**package** ExamPrep.DemoTechnologyFundamentalsMidExam02March2019;  
  
**import** java.util.Scanner;  
  
**public class** P1CookingMasterclass {  
 **public static void** main(String[] args) {  
 Scanner scanner = **new** Scanner(System.***in***);  
  
 **double** budget = Double.*parseDouble*(scanner.nextLine());  
 **int** students = Integer.*parseInt*(scanner.nextLine());  
 **double** flourPrice = Double.*parseDouble*(scanner.nextLine());  
 **double** eggPrice = Double.*parseDouble*(scanner.nextLine());  
 **double** apronPrice = Double.*parseDouble*(scanner.nextLine());  
  
 **int** freePackages = students / 5;  
 **double** neededMoney = apronPrice \* Math.*ceil*(students \* 1.2) +  
 eggPrice \* 10 \* (students) + flourPrice \* (students - freePackages);  
  
 **if** (neededMoney > budget) {  
 System.***out***.println(String.*format*(**"%.2f$ more needed."**, neededMoney - budget));  
 } **else** {  
 System.***out***.println(String.*format*(**"Items purchased for %.2f$."**, neededMoney));  
 }  
  
  
 }  
}

## 02. Bread Factory

**package** ExamPrep.DemoTechnologyFundamentalsMidExam02March2019;  
  
**import** java.util.Arrays;  
**import** java.util.Scanner;  
**import** java.util.stream.Collector;  
**import** java.util.stream.Collectors;  
  
**public class** P2BreadFactory {  
 **public static void** main(String[] args) {  
 Scanner scanner = **new** Scanner(System.***in***);  
  
 String[] events = scanner.nextLine().split(**"\\|"**);  
 String[] commands = **new** String[2];  
 **int** coins = 100;  
 **int** energy = 100;  
 **boolean** managed = **true**;  
 **for** (**int** i = 0; i < events.**length**; i++) {  
  
 commands = events[i].split(**"-"**);  
 **if** (commands[0].equals(**"rest"**)) {  
  
 **if** (energy == 100) {  
 System.***out***.println(**"You gained 0 energy."**);  
 } **else** {  
 energy += Integer.*parseInt*(commands[1]);  
 **if** (energy > 100) {  
 **int** restEnergy = energy - 100;  
 energy = 100;  
 System.***out***.printf(**"You gained %d energy.\n"**, Integer.*parseInt*(commands[1]) - restEnergy);  
  
 } **else** {  
 System.***out***.printf(**"You gained %d energy.\n"**, Integer.*parseInt*(commands[1]));  
 }  
 }  
 System.***out***.printf(**"Current energy: %d.\n"**, energy);  
 } **else if** (commands[0].equals(**"order"**)) {  
 **if** (energy >= 30) {  
 energy -= 30;  
 coins += Integer.*parseInt*(commands[1]);  
 System.***out***.println(String.*format*(**"You earned %d coins."**, Integer.*parseInt*(commands[1])));  
 } **else** {  
 energy += 50;  
 System.***out***.println(**"You had to rest!"**);  
 }  
  
 } **else** {  
 **if** (coins > Integer.*parseInt*(commands[1])) {  
 coins -= Integer.*parseInt*(commands[1]);  
 System.***out***.println(String.*format*(**"You bought %s."**, commands[0]));  
 } **else** {  
 System.***out***.println(String.*format*(**"Closed! Cannot afford %s."**, commands[0]));  
 **return**;  
 }  
  
 }  
  
  
 }  
 System.***out***.println(String.*format*(**"Day completed!\n"** +  
 **"Coins: %d\n"** +  
 **"Energy: %d\n"**, coins, energy));  
  
  
 }  
  
}

## 03. Cooking Factory

**package** ExamPrep.DemoTechnologyFundamentalsMidExam02March2019;  
  
**import** java.util.ArrayList;  
**import** java.util.Arrays;  
**import** java.util.List;  
**import** java.util.Scanner;  
**import** java.util.stream.Collectors;  
  
**public class** P3CookingFactory {  
 **public static void** main(String[] args) {  
 Scanner scanner = **new** Scanner(System.***in***);  
  
 List<Integer> bestBread = **new** ArrayList<>();  
 **int** bestSum = Integer.***MIN\_VALUE***;  
 String input = scanner.nextLine();  
 **while** (!input.equals(**"Bake It!"**)) {  
 List<Integer> current = Arrays.*stream*(input.split(**"\\#+"**))  
 .map(Integer::*parseInt*).collect(Collectors.*toList*());  
 **if** (current.size() > 10) {  
 **break**;  
 }  
 **int** currSum = current.stream().mapToInt(e -> e).sum();  
 **double** averageCurrent = (**double**) currSum / current.size();  
 **double** averageBest = (**double**) bestSum / bestBread.size();  
 **if** (currSum > bestSum) {  
 bestBread = current;  
 bestSum = currSum;  
 } **else if** (currSum == bestSum) {  
 **if** (averageCurrent > averageBest) {  
 bestBread = current;  
 bestSum = currSum;  
 } **else if** (averageCurrent == averageBest) {  
 **if** (current.size() < bestBread.size()) {  
 bestBread = current;  
 bestSum = currSum;  
 }  
 }  
 }  
 input = scanner.nextLine();  
 }  
  
 System.***out***.println(String.*format*(**"Best Batch quality: %d"**, bestSum));  
 bestBread.forEach(e -> System.***out***.print(e + **" "**));  
 }  
}